How to increase convergence order of Newton's method to $2 \times m$?

Sanjay Kumar Khattri [1]
Department of Engineering,
Stord Haugesund University College, Norway

Abstract: We present a simple yet powerful technique for forming iterative methods of various convergence orders. Methods of various convergence orders (four, six, eight and ten) are formed through a modest modification of the classical Newton method. The technique can be easily implemented in existing software packages as suggested by the presented C^{++} algorithm. Finally some problems are solved through the proposed algorithm.

1 Introduction

The most common and probably the most used method method finding a simple root γ, i.e. $f(\gamma) = 0$, of a nonlinear scalar equation

$$f(x) = 0, \tag{1}$$

is the Newton method. The classical Newton method is given as

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \qquad n = 0, 1, 2, 3, \ldots. \tag{2}$$

It is well documented and well known that the Newton's method converges quadratically [see 1; 2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 12; 14; 15; 16; 17; 18; 19; 20; 21; 22; 23; 24; 25; 26, and references therein]. There exists many modifications of the Newton method to improve convergence order [1; 2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 12; 14; 15; 16; 17; 18; 19; 20; 21; 22; 23; 24; 25; 26]. Higher order modifications of the Newton's method free of second or higher derivatives are actively researched. For example, third order convergent methods are presented in [16; 18; 20; 21], fourth order convergent methods are developed in [1; 2; 3;

[1]Corresponding author.
E-mail addresses: sanjay.khattri@hsh.no (S. K. Khattri)

4; 5; 6; 7; 8; 9; 10], sixth order methods are developed in [28; 29; 30] and eight order methods are presented in [see 31, and references therein]. As there exists various modifications of the Newtons method. And from practical point (implementing these methods into a software package) one of the drawbacks of these wonderful methods is their independent nature. For example, if one has a software package which solves nonlinear equations by the well known fourth order convergent Jarrat method [26]. One may find it difficult to modify the existing software package to implement sixth order methods [28; 29; 30] and eight order methods [31].

In this work, we develop a technique. Which improves the order of convergence of the Newton method (2) from 2 to $2 \times m$. Here, $m = 1, 2, 3, \ldots$. Thus through our scheme, one may develop 4th order, 6th order, 8th order, ... convergent iterative methods. One of the beautiful fact of our scheme is that one needs modest modifications in the classical iterative method (2) for achieving higher convergence rates. And, which may be very effective when one wants to modify an existing software package for achieving higher convergence order. Let us now develop our scheme.

2 The technique and convergence order of its various methods

Before presenting our technique, first we will develop iterative methods of various convergence orders. Let us consider the following 4th order convergent iterative method

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \tag{3}$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}\left[1 + \frac{f(y_n)}{f(x_n)}\left(1 + 2\,\frac{f(y_n)}{f(x_n)}\right)\right], \tag{4}$$

error equation for the above method is given as

$$e_{n+1} = -\frac{1}{12}\,\frac{c_2\left(12\,c_3c_1 - 60\,{c_2}^2\right)}{{c_1}^3}e_n^4 + O(e_n^5). \tag{5}$$

Here, $c_k = f^k(\gamma)/!k$ and $e_n = x_n - \gamma$. A proof of convergence of the above fourth order method is presented next.

proof

Using Taylor series of $f(x)$ around the solution γ, and taking into account $f(\gamma) = 0$, we get

$$f(x_n) = \sum_{k=1}^{\infty} c_k\, e_n^k, \tag{6}$$

furthermore from the equation (6) we have

$$f'(x_n) = \sum_{k=1}^{\infty} k\, c_k\ e_n^{k-1}, \tag{7}$$

and through a simple calculation we arrive at

$$\frac{f(x)}{f'(x)} = e_n - \frac{c_2}{c_1} {e_n}^2 - 2\,\frac{c_3 c_1 - {c_2}^2}{{c_1}^2} {e_n}^3 - \frac{3\,c_4 {c_1}^2 - 7\,c_2 c_3 c_1 + 4\,{c_2}^3}{{c_1}^3} {e_n}^4 + O\left({e_n}^5\right). \tag{8}$$

Substituting (8) in (3) yields

$$y_n - \gamma = \frac{c_2}{c_1} {e_n}^2 + 2\,\frac{c_3 c_1 + {c_2}^2}{{c_1}^2} {e_n}^3 + \frac{3\,c_4 {c_1}^2 + 7\,c_2 c_3 c_1 + 4\,{c_2}^3}{{c_1}^3} {e_n}^4 + O\left({e_n}^5\right). \tag{9}$$

Expanding $f(y_n)$ around the solution γ and using (9), we obtain

$$f(y_n) = c_2 {e_n}^2 - \frac{1}{6}\,\frac{-12\,c_3 c_1 + 12\,{c_2}^2}{c_1} {e_n}^3 + \frac{1}{24}\,\frac{72\,c_4 {c_1}^2 - 168\,c_2 c_3 c_1 + 120\,{c_2}^3}{{c_1}^2} {e_n}^4 + O\left({e_n}^5\right). \tag{10}$$

From equations (6) and (10), we get

$$\frac{f(y_n)}{f(x_n)} = \frac{c_2}{c_1} e_n + \frac{2\,c_3 c_1 - 3\,{c_2}^2}{{c_1}^2} {e_n}^2 - \frac{-3\,c_4 {c_1}^2 + 10\,c_2 c_3 c_1 - 8\,{c_2}^3}{{c_1}^3} {e_n}^3 + O\left({e_n}^4\right). \tag{11}$$

Now from equations (8), (11) and (4), we find that

$$e_{n+1} = -\frac{1}{12}\,\frac{c_2\left(-60\,{c_2}^2 + 12\,c_3 c_1\right)}{{c_1}^3} e_n^4 + O(e_n^5). \tag{12}$$

This proofs that the method (4) converges quartically.

Let us now consider the following three step sixth order convergent iterative method

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n &= x_n - \frac{f(x_n)}{f'(x_n)}\left[1 + \frac{f(y_n)}{f(x_n)}\left(1 + 2\,\frac{f(y_n)}{f(x_n)}\right)\right]. \\ x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)}\left[1 + \frac{f(y_n)}{f(x_n)}\left(1 + 2\,\frac{f(y_n)}{f(x_n)}\right)\right. \\ &\quad \left. + \frac{f(z_n)}{f(x_n)}\left(1 + 2\,\frac{f(y_n)}{f(x_n)}\right)\right], \end{aligned} \tag{13}$$

error equations for the above sixth order method is given as

$$e_{n+1} = \frac{1}{72}\,\frac{c_2\left(-792\,c_3c_1{c_2}^2 + 2160\,{c_2}^4 + 72\,{c_3}^2{c_1}^2\right)}{{c_1}^5}e_n^6 + O\left(e_n^7\right).$$

Convergence order of the above method can be easily established through the Maple software package. We may notice that the method (13) requires evaluations of only three functions and one derivative during each iterative step. Let us now further consider the following eight order convergent iterative step

$$\begin{aligned}
y_n &= x_n - \frac{f(x_n)}{f'(x_n)},\\
z_n &= x_n - \frac{f(x_n)}{f'(x_n)}\left[1 + \frac{f(y_n)}{f(x_n)}\left(1 + 2\,\frac{f(y_n)}{f(x_n)}\right)\right],\\
p_n &= x_n - \frac{f(x_n)}{f'(x_n)}\left[1 + \frac{f(y_n)}{f(x_n)}\left(1 + 2\,\frac{f(y_n)}{f(x_n)}\right) + \frac{f(z_n)}{f(x_n)}\left(1 + 2\,\frac{f(y_n)}{f(x_n)}\right)\right],\\
x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)}\left[1 + \frac{f(y_n)}{f(x_n)}\left(1 + 2\,\frac{f(y_n)}{f(x_n)}\right) + \frac{f(z_n)}{f(x_n)}\left(1 + 2\,\frac{f(y_n)}{f(x_n)}\right)\right.\\
&\quad\left. + \frac{f(p_n)}{f(x_n)}\left(1 + 2\,\frac{f(y_n)}{f(x_n)}\right)\right],
\end{aligned} \tag{14}$$

asymptotic error equation for the eight order method is given as

$$e_{n+1} = \frac{1}{432}\,\frac{c_2\left(77760\,{c_2}^6 - 41472\,c_3c_1{c_2}^4 + 7344\,{c_3}^2{c_1}^2{c_2}^2 - 432\,{c_3}^3{c_1}^3\right)}{{c_1}^7}e_n^8$$

We may notice that the eight order method (14) requires evaluations of only four functions and one derivative during each iterative step. Based upon the similarity in methods (4), (13) and (14). Let us consider the following method

$$\begin{aligned}
y_n &= x_n - \frac{f(x_n)}{f'(x_n)},\\
z_n &= x_n - \frac{f(x_n)}{f'(x_n)}\left[1 + \frac{f(y_n)}{f(x_n)}\left(1 + 2\,\frac{f(y_n)}{f(x_n)}\right)\right],\\
p_n &= x_n - \frac{f(x_n)}{f'(x_n)}\left[1 + \frac{f(y_n)}{f(x_n)}\left(1 + 2\,\frac{f(y_n)}{f(x_n)}\right) + \frac{f(z_n)}{f(x_n)}\left(1 + 2\,\frac{f(y_n)}{f(x_n)}\right)\right],\\
q_n &= x_n - \frac{f(x_n)}{f'(x_n)}\left[1 + \frac{f(y_n)}{f(x_n)}\left(1 + 2\,\frac{f(y_n)}{f(x_n)}\right) + \frac{f(z_n)}{f(x_n)}\left(1 + 2\,\frac{f(y_n)}{f(x_n)}\right)\right.\\
&\quad\left. + \frac{f(p_n)}{f(x_n)}\left(1 + 2\,\frac{f(y_n)}{f(x_n)}\right)\right],\\
x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)}\left[1 + \frac{f(y_n)}{f(x_n)}\left(1 + 2\,\frac{f(y_n)}{f(x_n)}\right) + \frac{f(z_n)}{f(x_n)}\left(1 + 2\,\frac{f(y_n)}{f(x_n)}\right)\right.\\
&\quad\left. + \frac{f(p_n)}{f(x_n)}\left(1 + 2\,\frac{f(y_n)}{f(x_n)}\right) + \frac{f(q_n)}{f(x_n)}\left(1 + 2\,\frac{f(y_n)}{f(x_n)}\right)\right],
\end{aligned} \tag{15}$$

through the Maple we verified that the above method is 10th order convergent, and error equation for it is given as

$$e_{n+1} = \frac{c_2}{2592\,c_1^9}\left(2799360\,{c_2}^8 - 1959552\,c_3c_1{c_2}^6 + 513216\,{c_3}^2{c_1}^2{c_2}^4 \right.$$
$$\left. -59616\,{c_3}^3{c_1}^3{c_2}^2 + 2592\,{c_3}^4{c_1}^4\right) e_n^{10}.$$

We may notice that the above tenth order method (15) requires evaluations of only five functions and one derivative per iterative step.

Based upon the methods (4), (13), (14) and (15), we conjuncture the existence of the following scheme for generating iterative method of order $2 \times m$

$$\begin{aligned}
y_1 =& x_n - \frac{f(x_n)}{f'(x_n)},\\
y_2 =& x - \frac{f(x_n)}{f'(x_n)}\left[1 + \frac{f(y_1)}{f(x_n)}\left(1 + 2\frac{f(y_1)}{f(x_n)}\right)\right],\\
y_3 =& x_n - \frac{f(x_n)}{f'(x_n)}\left[1 + \frac{f(y_1)}{f(x_n)}\left(1 + 2\frac{f(y_1)}{f(x_n)}\right) + \frac{f(y_2)}{f(x_n)}\left(1 + 2\frac{f(y_1)}{f(x_n)}\right)\right],\\
y_4 =& x_n - \frac{f(x_n)}{f'(x_n)}\left[1 + \frac{f(y_1)}{f(x_n)}\left(1 + 2\frac{f(y_1)}{f(x_n)}\right) + \frac{f(y_2)}{f(x_n)}\left(1 + 2\frac{f(y_1)}{f(x_n)}\right) + \frac{f(y_3)}{f(x)}\left(1 + 2\frac{f(y_1)}{f(x_n)}\right)\right],\\
&\vdots\\
y_{m-1} =& x_n - \frac{f(x_n)}{f'(x_n)}\left[1 + \frac{f(y_1)}{f(x_n)}\left(1 + 2\frac{f(y_1)}{f(x_n)}\right) + \cdots + \frac{f(y_{m-2})}{f(x_n)}\left(1 + 2\frac{f(y_1)}{f(x_n)}\right)\right],\\
x_{n+1} =& x_n - \frac{f(x_n)}{f'(x_n)}\left[1 + \frac{f(y_1)}{f(x_n)}\left(1 + 2\frac{f(y_1)}{f(x_n)}\right) + \cdots + \frac{f(y_{m-1})}{f(x_n)}\left(1 + 2\frac{f(y_1)}{f(x_n)}\right)\right].
\end{aligned} \tag{16}$$

It may be notice that a $2 \times m$ order method formed by the above scheme will require m functions and one deravitive evaluation per iterative step. A C^{++} implementation of the above scheme (16) is presented in the Listing 1.

3 Numerical work

The order of convergence ξ of an iterative method is defined as follows

$$\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^\xi} = c \neq 0. \tag{17}$$

Here, e_n is the error after n iterations of the method. Then the approximate value of the computational order of convergence (COC) ρ [27] can be find using the formula

$$\rho \approx \frac{\mathrm{Log}|(x_{n+1} - \gamma)/(x_n - \gamma)|}{\mathrm{Log}|(x_n - \gamma)/(x_{n-1} - \gamma)|}.$$

Listing 1: C++ implementation.

```
int main(){
    mp::mp_init(2005);
    mp_real tol =  1.0e-300;
    unsigned int w = 5;
    unsigned int maxitr = 1000;
    std::vector<mp_real> x(maxitr,0.0);
    x[0] = -0.5;
    mp_real err1 = 100.0;
    mp_real err2 = 100.0;
    mp_real gamma = -1.0;
    unsigned int m = 1;
    unsigned i = 0;
    unsigned int conv = 0;
    while(err1 > tol || err2 > tol){
        if(i >= maxitr){
            break;
        }
        x[i+1] = x[i] - function(x[i])/firstderivative(x[i]);
        mp_real x1 = x[i+1];
        mp_real x0 = x[i];
        for(unsigned int k = 1 ; k < m ; ++k){
            mp_real y = x[i+1];
            x[i+1] = x[i+1] - function(x[i+1])
                      /firstderivative(x0)*(1.0
                            + 2.0 * function(x1)/function(x0));
            err1 = abs(y-x[i+1]);
            err2 = abs(function(x[i+1]));
            if(err1 < tol & err2 < tol){
                std::cout << "err1 = " << scientific
                          << setw(2*w) << err1
                          << "     err2 = " <<setw(2*w)
                          << err2 <<  std::endl;
                conv = 1;
                break;
            }
        }
        if(conv) break;
        err1 = abs(x[i+1] - x[i]);
        err2 = abs(function(x[i+1]));
        std::cout << setw(w)  << "itr.=" << i << scientific
                  << "    " << setw(2*w) << x[i+1]
                  << std::endl;
        ++i;
    }
    mp_real rho;
    for (unsigned int n = 1; n < i; ++n){
        rho = log(abs( (x[n+1] - gamma)/(x[n]-gamma)))
                 /log(abs( (x[n] - gamma)/(x[n-1]-gamma)));
        std::cout << "rho = " << rho << std::endl;
    }
    mp::mp_finalize();
    return 1;
}
```

All the computations reported here are done in the programming language C^{++}. For numerical precision, we are using ARPREC[32]. The ARPREC package supports arbitrarily high level of numeric precision[32]. In the program (see the Listing 1) the precision in decimal digits is set at 2005 with the command mp::mp init(2005)[32]. We have implemented the presented technique in the C^{++} language. Listing 1 presents the main part of our implementation.

For convergence, it is required that the distance of two consecutive approximations ($|x_{n+1} - x_n|$) be less than ϵ. And, the absolute value of the function ($|f(x_n)|$) also referred to as residual be less than ϵ. Apart from the convergence criteria, our algorithm also uses maximum allowed iterations as stopping criterion. Thus our algorithm stops if (i) $|x_{n+1} - x_n| < \epsilon$ (ii) $|f(x_n)| < \epsilon$ (iii) itr $>$ maxitr. Here, $\epsilon = 1 \times 10^{-300}$, itr is the iteration counter for the algorithm and maxitr $= 100$. See the C^{++} algorithm presented in the Listing 1. The algorithm is tested for the following functions [31]

$$\begin{aligned}
f_1(x) =& x^5 + x^4 + 4x^2 - 15, & \gamma \approx&\ \ 1.347,\\
f_2(x) =& \sin x - x/3, & \gamma \approx&\ \ 2.278,\\
f_3(x) =& 10x\, e^{-x^2} - 1, & \gamma \approx&\ \ 1.679,\\
f_4(x) =& \cos x - x, & \gamma \approx&\ \ 0.739,\\
f_5(x) =& e^{-x^2+x+2} - 1, & \gamma \approx& -1.000,\\
f_6(x) =& e^{-x} + \cos x, & \gamma \approx&\ \ 1.746,\\
f_7(x) =& \mathrm{Log}(x^2 + x + 2) - x + 1, & \gamma \approx&\ \ 4.152,\\
f_8(x) =& \arcsin(x^2 - 1) - x/2 + 1, & \gamma \approx&\ 0.5948.
\end{aligned}$$

We run the algorithm shown in the Listing 1 for four values of m: $m = 1, 2, 3, 4$. Here, $m = 1$ corresponds to the classical Newton method. We choose the same initial guess as found in the article [31]. So the reader may find it easier to compare performance of various methods. Table 1 reports outcome of our numerical work. The table 1 reports (iterations required, number of function evaluations needed, COC during second last iteration) for the Newton method ($m = 1$), fourth order iterative method ($m = 2$), sixth order iterative method ($m = 3$) and eight order iterative method ($m = 4$). Computational order of convergence reported in the Table 1 was observed during the second last iteration. Following important observations were made during numerical experimentations

1. In the Table 1, the methods which require least number of function evaluations for convergence are marked in bold.

2. As reported in the Table 1, for the function $f_2(x)$, COC (during the second last iteration) is same for $m = 1, 2, 3, 4$. While COC for the second, third and fourth iterations is reported in the Table 2.

3. From the Table 1, we notice that for the functions $f_3(x)$, $f_4(x)$, $f_7(x)$ and $f_8(x)$ the sixth order ($m = 3$) and eight ($m = 4$) methods requirs same number of iterative steps. Table 3 reports residual ($|f(x_n)|$) during the last iterative step.

$f(x)$	x_0	NM($m=1$)	$m=2$	$m=3$	$m=4$
$f_1(x)$	1.6	$(9,18,2)$	$(4,12,4)$	$(2,\mathbf{8},5.66)$	$(2,10,7.6)$
$f_2(x)$	2.0	$(23,46,1)$	$(10,30,1)$	$(7,\mathbf{21},1)$	$(6,30,1)$
$f_3(x)$	1.8	$(10,20,2)$	$(4,\mathbf{12},3.99)$	$(3,\mathbf{12},6.21)$	$(3,15,8.22)$
$f_4(x)$	1.0	$(9,18,2)$	$(4,\mathbf{12},3.99)$	$(3,\mathbf{12},5.90)$	$(3,15,8.10)$
$f_5(x)$	-0.5	$(11,22,2)$	$(5,\mathbf{15},3.99)$	$(4,16,5.99)$	$(3,\mathbf{15},6.75)$
$f_6(x)$	2.0	$(9,18,2)$	$(4,12,3.99)$	$(3,12,5.99)$	$(2,\mathbf{10},8.10)$
$f_7(x)$	3.2	$(10,20,2)$	$(4,\mathbf{12},3.99)$	$(3,\mathbf{12},6.19)$	$(3,15,8.19)$
$f_8(x)$	1.0	$(10,20,2)$	$(4,\mathbf{12},4.01)$	$(3,\mathbf{12},6.35)$	$(3,15,8.36)$

Table 1: (iterations, number of function evaluations, COC) for the Newton method ($m = 1$), fourth order method ($m = 2$), sixth order method ($m = 3$) and eight order iterative method ($m = 4$).

$f(x)$	$m=1$	$m=2$	$m=3$	$m=4$
$f_2(x)$	10^{-435}	10^{-872}	10^{-671}	10^{-1522}

Table 2: Computational order of convergence (COC) at the second iteration.

$f(x)$	$m=1$	$m=2$	$m=3$	$m=4$
$f_7(x)$	10^{-435}	10^{-872}	10^{-671}	10^{-1522}
$f_8(x)$	10^{-347}	10^{-744}	10^{-800}	10^{-1302}

Table 3: Residual ($|f(x_n)|$)

References

[1] J. Kou, Y. Li, X. Wang, A composite fourth-order iterative method for solving non-linear equations, Appl. Math. Comput. 184 (2007) 471-475.

[2] C. Chun, Y. Ham, A one-parameter fourth-order family of iterative methods for nonlinear equations, Appl. Math. Comput. 189 (2007) 610-614.

[3] J.R. Sharma and R.K. Goyal, Fourth-order derivative-free methods for solving non-linear equations, Int. J. Comput. Math. 83 (2006) 101–106.

[4] C. Chun, Some fourth-order iterative methods for solving nonlinear equations, Appl. Math. Comput. 195 (2008) 454-459.

[5] M.A. Noor, F. Ahmad, Fourth-order convergent iterative method for nonlinear equation, Appl. Math. Comput. (2006), doi:10.1016/j.amc.2006.04.068.

[6] J. Kou, Y. Li, X. Wang, Fourth-order iterative methods free from second derivative, Appl. Math. Comput. (2006), doi:10.1016/j.amc.2006.05.189.

[7] R. King, A family of fourth-order methods for nonlinear equations, SIAM J. Numer. Anal. 10 (1973) 876–879.

[8] C. Chun, A family of composite fourth-order iterative methods for solving nonlinear equations, Appl. Math. Comput.187 (2007) 951-956.

[9] I.K. Argyros, D. Chen and Q. Qian, The Jarratt method in Banach space setting, J. Comput. Appl. Math. 51 (1994), 1-3.

[10] C. Chun, Y. Ham, Some fourth-order modifications of Newton's method, Appl. Math. Comput. 197 (2008) 654-658.

[11] J. Kou, Y. Li, X. Wang, Fourth-order iterative methods free from second derivative, Appl. Math. Comput. 184 (2007) 880-885.

[12] A. K. Maheshwari, A fourth order iterative method for solving nonlinear equations, Appl. Math. Comput. In Press, Accepted Manuscript, Available online 28 January 2009.

[13] X. Feng, Y. He, High order iterative methods without derivatives for solving nonlinear equations, Appl. Math. Comput. 186 (2007) 1617-1623.

[14] S. Abbasbandy, Modified homotopy perturbation method for nonlinear equations and comparison with Adomian decomposition method, Appl. Math. Comput. 172 (2006) 431-438.

[15] S. Weerakoon and T.G.I. Fernando, A Variant of Newton's Method with Accelerated Third-Order Convergence, Appl. Math. Lett. 13 (2000) 87-93.

[16] H.H. Homeier, On Newton-type methods with cubic convergence, J. Comput. Appl. Math. 176 (2005) 425-432.

[17] F.A. Potra and V. Pták, Nondiscrete induction and iterative processes, Research Notes in Mathematics vol. 103, Pitman, Boston (1984).

[18] J. Kou, Y. Li, X. Wang, A modification of Newton method with third-order convergence, Appl. Math. Comput. 181 (2006) 1106-1111.

[19] S. K. Khattri, Altered Jacobian Newton Iterative Method for Nonlinear Elliptic Problems. IAENG International Journal of Applied Mathematics, 38:3, IJAM_38_3_01, 2008.

[20] M. Frontini and E. Sormani, Some variant of Newton's method with third-order convergence, Appl. Math. Comput. 140 (2003) 419–426.

[21] A.Y. Özban, Some new variants of Newton's method, Appl. Math. Lett. 17 (2004) 677–682.

[22] S. K. Khattri, Newton-Krylov Algorithm with Adaptive Error Correction for the Poisson-Boltzmann Equation, MATCH Commun. Math. Comput. Chem. 56 (2006) 197–208.

[23] C. Chun, A geometric construction of iterative functions of order three to solve nonlinear equations, Comput. Math. Appl. 53 (2007) 972-976.

[24] A.M. Ostrowski, Solutions of Equations and System Equations, Academic Press, New Youk, 1960.

[25] J. F. Traub. Iterative methods for the solution of equations, Chelsea Publishing Company, New Youk, 1977.

[26] I. K. argyros, D. Chen, Q. Qian, The Jarrat method in Banach space setting, J. Computing. Appl. Math. 51 (1994).

[27] Changbum Chun. Construction of Newton-like iteration methods for solving nonlinear equations. Numerische Mathematik. Volume 104, Number 3 / September, 2006.

[28] C. Chun, Y. Ham. Some sixth-order variants of Ostrowski root-finding methods. Appl. Math. Comput. 193, 2003

[29] H. Ren, Q. Wu and W. Bi. New variants of Jarratts method with sixth-order convergence. Numerical Algorithms, Volume 52, Number 4, December, 2009.

[30] J.R. Sharma, and R.K. Guha. A family of modified Ostrowski methods with accelerated sixth order convergence. Appl. Math. Comput. 190, 2007.

[31] Weihong Bi, Hongmin Ren and Qingbiao Wu. Three-step iterative methods with eighth-order convergence for solving nonlinear equations. Journal of Computational and Applied Mathematics, Volume 225, Issue 1, 1 March 2009, Pages 105-112.

[32] ARPREC. C++/Fortran-90 arbitrary precision package. Available at `http://crd.lbl.gov/~dhbailey/mpdist/`.